1995

Speech discrimination using wavelets and zero crossing counting

Craig Martin

Speech Discrimination using Wavelets and Zero Crossing Counting

by

Craig W. Martin

M.S. Rochester Institute of Technology

A Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in the School of Computer Science in the College of Applied Science and Technology of the Rochester Institute of Technology.

May, 1995

College of Applied Science and Technology
Rochester Institute of Technology
Rochester, New York

CERTIFICATE OF APPROVAL

M. S. DEGREE THESIS

The M. S. Thesis of Craig W. Martin
has been examined and approved
by the thesis committee as satisfactory
for the thesis requirement for the
Master of Science Degree

Dr. P.G. Anderson, Thesis Advisor

Dr. J. A. Biles

Dr. M. Raghuveer

15 May 1995

Date

THESIS RELEASE PERMISSION FORM
ROCHESTER INSTITUTE OF TECHNOLOGY
COLLEGE OF APPLIED SCIENCE AND TECHNOLOGY

Title of Thesis: Speech Discrimination using Wavelets and Zero Crossing Counting

I, Craig Martin, prefer to be contacted each time a request for reproduction is made. I can be reached at the following address:

Eastman Kodak Company

Research Laboratories, B-59

Rochester, New York, 14560

Date: May 1, 1995

SPEECH DISCRIMINATION USING WAVELETS AND ZERO CROSSING COUNTING

by

Craig W. Martin

Submitted to the
Computer Science Department
in partial fulfillment of the requirements
for the Master of Science degree
at the Rochester Institute of Technology

ABSTRACT

Traditional speech discrimination techniques have focused primarily on the frequency domain analysis of speech signals. In this paper the author presents a new technique using wavelet transforms and zero crossing counting to facilitate a time domain speech discrimination system.

ACKNOWLEDGMENTS

This work would not have been possible without the significant contributions made by a number of kind individuals, whom I gratefully acknowledge:

Dr. P. G. Anderson, for his patience, time and helpful suggestions in serving as my thesis advisor.

Dr. J. A. Biles, for his time and suggestions in reviewing this thesis.

Dr. M. Raghuveer, for his time and suggestions in reviewing this thesis.

Dr. B. W. Keelan, for his assistance in applying the SAS discrimination capabilities to speech recognition.

and the members of Eastman Kodak department 791 for helping me to populate my database of Petersen/Barney sounds.

DEDICATION

This thesis is dedicated to my lovely wife, Nancy, and my two children, Andrew and Kristy, whose undying patience made this thesis possible.

CONTENTS

INTRODUCTION

Many techniques are currently used in pursuit of the perfect speech recognition system. These techniques encompass various levels of recognition, signal processing, and comprehension. Most of these systems begin with digitally sampled speech followed by some sort of signal processing. The signal processing may encompass spectral analysis, LPC analysis, Fourier and Gabor transforms, and various other techniques [1-7]. In addition, other systems attempt to understand speech by applying artificial intelligence techniques to the speech signal. Most of the more traditional speech recognition techniques have their foundations rooted in the frequency domain analysis of speech signals. Overviews of this type of system have been widely published [1] and generally follow these basic steps:

(1) Find the beginning and end of the utterance.

(2) Filter the raw signal into frequency bands (FFT, Gabor, etc.).

(3) Cut the utterance into a fixed number of segments.

(4) Average the data for each band in each segment.

(5) Store this pattern with its name.

(6) Collect a training set of about 3 repetitions for each pattern.

(7) Recognize the unknown by comparing its pattern against all patterns in the training set and returning the name of the pattern closest to the unknown.

Much success has been had with different variations of this simple speech recognition system.

The success of the frequency domain approach to speech recognition is in no small part due to the fact that this type of analysis eliminates the need for phase detection algorithms. These algorithms, which are generally quite complex [8], are required when comparing speech or other signals in the time domain. This is the best way to obtain a common reference point for the comparison. The phase detection issue has in part been responsible for the lack of speech recognition work using time domain techniques.

It is not surprising then that the wavelet transform, which generally produces coefficients with heavy dependence on temporal positioning (at higher frequencies), has been used sparingly in speech recognition work. In spite of this, recent research [9,13] has shown the following advantages in using wavelets to solve the speech recognition problem:

(1) Contained in the wavelet transform is information pertaining to both frequency content (harmonic) and the regions of energy concentration that constitute the frequency spectrum of a speech signal (formant).

(2) The analysis process used by the wavelet transform is very similar to that performed by the human ear.

(3) Due to the time localization properties of the wavelet transform, it is quite valuable for locating discontinuities in speech signals. This is particularly helpful for locating the beginning and ending points of vowels.

(4) The wavelet transform has the property of compressing most of the target signal properties into a few of the largest coefficients. This minimizes the amount of data associated with the speech signal exemplars.

This thesis is based on the hope that the aforementioned wavelet properties can be exploited to produce a simple vowel discrimination system based on information extracted from the wavelet transforms. As an additional goal, the system should not be so computationally complex as to require super computer capabilities to run the algorithm. The feature extraction algorithm should be able to be run, with minimal delay, on standard PC/Mac systems.

Tasks not pursued in this thesis are complete word/sentence recognition, vowel discrimination outside the nine Petersen/Barney vowels, or foreign pronunciations or accent modifications of the nine Petersen/Barney vowels. These projects would be logical extensions of this work.

Defining and Populating the Vocabulary Database

Before any work could begin on this project it was incumbent on us to define and populate a database of speech on which the wavelet based recognition system would operate. Early on in this project the decision was made to use the Petersen/Barney group of vowels to populate the database. These are a group of nine internationally recognized vowels which, for the most part, do not overlap in frequency space. This decision was based on two factors:

(1) The wavelet transform has been shown to be especially well suited to detecting the onset and offset of vowels [9,13].

(2) Having entered this project *a priori*, this choice limited the scope of the database in terms of the expectations placed on the speech recognition system.

Once the scope of the database was defined, and before any real work could commence, a plan for populating the speech database with real speech samples was devised and implemented.

The first step in population of the database was to find a willing group of speakers who would provide a wide number of variations of each of the Petersen/Barney vowels. Fortunately, I was able to solicit volunteers for this purpose from a number of people working with me at Eastman Kodak. This group was made up of approximately twelve people who formed the foundation of the database from the beginning of the project. I am deeply grateful for their help.

The next step was to devise a system which would aid in both correct vowel pronunciation by the participants and efficient software access to the vowels contained in the database. After careful examination of the problem, a series of nine words were chosen, each beginning with one of the vowels in the Petersen/Barney group. This vocabulary is shown in the table below:

Figure 0

Sound #	Pronunciation Word	Petersen/Barney Vowel
1	eat	$\bar{e}$
2	it	i
3	any	e
4	at	a
5	up	u
6	Amish	ä
7	oomph	o͝o
8	ooze	o͞o
9	auto	ô

In having each participant say each of these nine words, we gain the advantage of aiding in proper pronunciation, and allowing the vowels to be easily parsed from the front of each word by a simple software algorithm (the parsing algorithm is discussed later in this paper). Thus the thesis vocabulary database consists of pronunciations of each of these nine words for each one of the 12 participants.

The final step was to find a dependable and consistent vehicle for recording and storing the sounds for the vocabulary database. Towards this end a Macintosh Quadra 650 computer was used. All database sounds were recorded on a single Quadra system using the same microphone (Sun SPARC 10 Microphone) each time. The Quadra 650 system was extremely easy to use. Some of the features which were helpful in the database effort were:

(1) The Macintosh presents a user friendly interface via the Sound Control Panel which is part of the Mac operating system. The Sound Control Panel contains simple buttons for controlling all aspects of recording and playback. This simplicity allowed sounds to be recorded and verified in very little time.

(2) All aspects of the resulting sound files are well documented by Apple Computer [11,12] (see the next section). This made writing software to operate on the sound data straightforward.

(3) The resulting sound files are compatible with any other Macintosh computer allowing the feature extraction software to be debugged and tested on other Macintosh systems. This includes systems with no sound recording capabilities.

The Macintosh Sound System

The purpose of this section is not to provide an exhaustive explanation of the Macintosh sound system. Apple Computer [11,12] has already addressed this issue. Rather, this section is intended to highlight the areas that have some direct influence on this thesis work. The main question that needs to be resolved is whether or not speech recorded on the Macintosh meets the minimum requirements for a speech discrimination system.

The first area that needs to be examined is that of the sampling rate used in the speech recording process. In general, it is agreed that a minimum sampling rate of 8kHz is required [1]. This allows speech with harmonics up to about 4kHz to be accurately represented. This assumes that other problems, such as aliasing, are not a major factor. As the sampling rate is increased, the upper frequency limit also increases. In general, the maximum frequency response is approximately equal to one half the sampling rate. The sampling rate used by the Macintosh computers in this thesis work is 22kHz. Thus, we have exceeded the minimum frequency response recommendation and have no concerns in this area.

The second important issue is to determine the number of bits required to represent each sampled point in the sound. Current research shows that a minimum of 8 bits per sample is generally recommended for intelligible speech [1]. By increasing the number of bits per sample, an improvement in the signal to noise ratio can be realized. The Macintosh systems used in this work employ an 8 bit representation where the data is stored as numbers ranging from 0 to 255. The value zero represents the largest negative amplitude and 255 represents the largest positive amplitude. A value of 127 is interpreted as an amplitude of 0. In this format, we have met the minimum required number of bits per sample. Most of the newer Macintosh computers now use a 16 bit representation which could yield improved results in future speech work.

The third and final issue is that of the frequency response of the microphone. The microphone shipped by Sun Microsystems with the Sun SPARC station 10 has a frequency response of 50Hz to 8kHz [14]. This would suggest that this is not a limiting factor since we determined previously that an upper frequency response limit of 4kHz was sufficient.

Speech Signal Preparation

In most speech recognition applications some form of pre-processing is done prior to the actual speech discrimination process. This signal preparation stage usually includes signal detection (lead and trailing edge), amplitude normalization, and phase normalization. This project is no different. In this case both lead/trail signal detection and amplitude normalization techniques were employed prior to doing any wavelet based feature extraction work. In addition, a simple algorithm was used to provide a crude estimation of the pitch. Phase normalization, which is required because of the time domain nature of wavelets, was not done in the preparation stage. This is due to the fact that a form of phase detection is built into the wavelet stage. This phase detection technique is one of the unique features of this work and will be discussed later in this paper.

As part of this thesis, a Macintosh program was written which handles the job of extracting exemplars from each sound and storing them in a file. In the first portion of this task, the program had to detect the lead edge of each vowel from the group of nine words discussed above, parse it in its entirety from each word, estimate its pitch, and perform amplitude normalization prior to the wavelet stage.

Signal detection algorithms are well documented [1,8] and can be quite complex. In the case of this work, I was able to use the fact that each of the nine words begin with the vowel in question, and exploit this to simplify the algorithm. Thus, in keeping with the originally stated goals of minimizing computational complexity, a simple vowel detection algorithm was developed by empirical means and verified. The main function of the algorithm was to throw away any noise spikes (such as lip smacks or breaths) occurring at the start of the vowel, and then begin buffering the data until a 1024 byte buffer was full. The algorithm operates directly on the 8 bit data discussed in a previous section and was implemented as follows:

> Step 1: Begin reading sound data until the difference between the data and zero amplitude (127 in this case) is greater than 3.
> Step 2: Read a data point and place it in the 1024 byte buffer.
> Step 3: Determine if 10 consecutive points have been read with an absolute amplitude less than 4. If the answer is yes, go to step 1 and restart from the current location.
> Step 4: Determine if 1024 bytes have been read and buffered. If the answer is no, go to step 2.
> Step 5: Exit the signal detection algorithm.

Two important assumptions were made here in order to simplify the algorithm. Firstly, the occurrence of ten values very close to zero indicates the end of a noise spike. Secondly, 1024 bytes of data are sufficient to capture the entire vowel at the beginning of each word.

The empirical process by which these two assumptions were verified was really quite straightforward. A menu item called "Play Sound Segment" was added to the exemplar extraction program written for the Macintosh. Using this feature in conjunction with adjustments to the number of "noise zeros" and the buffer size, the vowel extractor was optimized. The optimization process was performed on all nine vowels for 12 different speakers. By making adjustments and then playing the extracted vowel, the values of 10 "noise zeros" and the 1024 byte buffer size were derived.

Pitch is very helpful in speech discrimination in that it can provide an additional piece of information which may aid in the comparison of multiple speech signals. Pitch algorithms, like speech signal detection, can be very complex [1]. But there are short cuts that can be used to provide "rough" estimates of pitch at a low cost in CPU cycles. One such short cut is to simply count the number of zero crossings in the utterance. Toward this end a trivial subroutine was written to count the number of zero crossings in the parsed vowel. This was done by incrementing a counter if one of the following conditions was met:

(1) A point that is less than or equal to zero is followed by a point greater than zero.

(2) A point that is greater than or equal to zero is followed by a point less than zero.

The importance of the pitch estimate will become clear later in this paper.

The last step before the wavelet stage was to perform amplitude normalization on the parsed vowel. These algorithms are fairly basic and also readily available [1,8]. In short, normalization is nothing more than multiplying every Fourier coefficient (frequency domain) or every sampled data point (time domain) by a pre-determined constant. In this case, all time domain data points were divided by the largest data point in the 1024 byte vowel buffer. This has the consequence of producing a signal with a maximum amplitude of one and a minimum amplitude of zero. There are several advantages resulting from this action:

(1) Since the coefficients resulting from wavelet transforms are somewhat dependent on input signal amplitude, it is desirable that all input data cover the same amplitude range. This forces the resulting coefficients to be as close as possible when trying to compare similar sounds.

(2) For the purpose of memory conservation and execution speed, four-byte floating point variables were used in the wavelet algorithm. Normalizing the input data to one prevented floating point overflow during these calculations.

(3) Dividing all data points by a constant has the possible effect of suppressing any noise component remaining in the signal. Multiplying by a constant greater than one may have resulted in exaggerating the noise and making speech discrimination more difficult.

(4) Normalization is a simple procedure in keeping with the original goal of avoiding mathematical procedures which are overly complex.

A simple subroutine was written to perform the above normalization algorithm. After parsing and normalization, the vowel is ready for the wavelet processing stage. Before getting into the specific details of how this was applied to this project, some of the theory behind the wavelet transform should be reviewed. The next section is devoted to this topic.

Wavelet Theory and Practice

In its basic form the wavelet transform is defined as:

$$W(\tau,\partial) = (1/\sqrt{\partial}) \int h((t-\tau)/\partial)\, x(t)\, dt \qquad [8,10,13]$$

where integration covers the range of plus and minus infinity. The wavelet, $h(t-\tau/\partial)$, defines a group of functions with variable temporal and frequency localization dependent on the scaling factor ∂. Each of these functions is actually a bandpass filter which is sensitive to events occurring around time τ and at a scale (frequency range) of ∂. The time and scale parameters can be varied to describe a family of wavelet functions described as a class. The shape of every member of the class is identical to every other member except for the scale change. These members are all extracted from one mother function by using the scale factor to expand or compress the end points.

The wavelet transform can be described as a constant Q analysis method. The ratio of bandwidth to center frequency remains constant which means that it enhances frequency resolution at lower frequencies and time resolution at higher frequencies. Thus, the inherent beauty of the wavelet is its ability to focus its high frequency analysis at a specific point in time, much like the analysis performed by the human ear. As mentioned in the introduction, this characteristic makes the wavelet well adapted to detecting the onset and offset of vowels and other speech discontinuities.

Even though a multitude of wavelet functions (covering the entire frequency spectrum) can be derived from a suitable mother wavelet, this is of limited value in the real world. The speech discrimination algorithm, whether it be linear or non-linear in nature, can only process exemplars which have a limited number of data points. Thus, we would like to adjust the values of τ and ∂ in order to limit the continuous domain of wavelets to a sparse set of functions. One such possibility is to choose a specialized set of values for τ and ∂ which approximate the Q of the human ear. This approach, which has been investigated by P. Basile, F. Cutugno, P. Maturi, and A. Piccialli [13], was beyond the limits of computational complexity (CPU cycles) imposed on this project, but showed definite promise. In addition, an investigation into adaptive wavelets by Harold H. Szu, Brian Telfer, Shubha Kadambe, and Pramila Srinivasan [16,17] was shown to produce very low speech classification errors. In this effort, Daubechies wavelet parameters were iterated in conjunction with a neural network to produce a minimum energy function in each band. This iteration was performed on a single pitch period for a given vowel. The adaptive wavelet approximation was then treated as a super wavelet which can be dilated and contracted to handle pitch and speaking rate changes. Once again, this was beyond the CPU cycle limit imposed on this work.

In the case of this thesis work, an existing wavelet algorithm was sought, which would provide a good tradeoff between CPU overhead and the ability to extract meaningful exemplars from the Petersen/Barney vowels.

Through the course of this thesis, the Daubechies class of wavelet filters [9,10,18] was found (by empirical means) to provide an acceptable tradeoff between CPU execution time, exemplar size presented to the discrimination algorithm, and the resulting discrimination rate. This judgment was made based on target exemplar extraction times in seconds (instead of minutes) with expected minimum discrimination rates of approximately 75%. These tests were run on an 8 MHz Macintosh SE with the assumption that performance on today's improved desktop systems would be orders of magnitude better.

The members of the Daubechies class vary from being highly localized in frequency scale to highly smooth depending on the size of wavelet filter convolution matrix. The simplest and most compact member of this class is DAUB4 and uses only 4 of these matrix values. In spite of the highly localized nature of this member, it was used in this thesis as the main vehicle for exemplar extraction. The advantage of this decision was found in the straightforwardness of the algorithm and the knowledge that success would bode well for results using more complex wavelets.

Numerical Recipes in C [9,10], presents algorithms in C for the discrete Daubechies wavelet transforms. These are given for convolution matrix sizes of 4, 12, and 20 values. Again, these impact the accuracy of the filtering. This C code was used with minor modification for this project. The transform operates on vectors which are a power of two in length (128, 256, 512, etc.) and in this case contain the discrete sound samples after signal processing (parsing, normalization, etc.). After the wavelet transform, the vector is returned as a series of coefficients, with emphasis on temporal placement of these coefficients increasing with frequency. The resulting coefficients are grouped by powers of two starting with the lowest frequencies. Thus, after transforming a 512 byte input vector, the first two coefficients (vector locations 0 and 1) represent analysis of the lowest frequencies, followed by the next 2, the next 4, the next 8, the next 16, ..., with the last 256 bytes representing analysis of the highest frequencies.

It should be obvious from this progression that there is increasing resolution, with respect to time, in the placement of these output coefficients as the target frequency band increases. This is because the bandpass filter represented by each successive member of the wavelet class has a wider scope than its predecessor while retaining the shape specified by τ and ∂ (shape specific to the Daubechies class). Furthermore, the speech signal can be quite accurately reconstructed after zeroing out all but the few largest coefficients in each band, and performing an inverse wavelet transform. The location of these coefficients in each band must also be retained. It is this ability to "compress" most of the relevant information into a few of the resulting coefficients that makes the wavelet transform valuable in many of today's image compression schemes. This capability for data compression has also been exploited here, to minimize the size of the exemplars extracted from the Petersen/Barney vowel set. More detail on this process will be provided in the next section.

Exemplar Extraction

The goal in the exemplar extraction process is to pull out a few key pieces of information that will demonstrate the unique features of the item in question. If possible it is best to keep the quantity of data in each exemplar to a minimum in order to lessen the burden on the discrimination algorithm. In the case of the Petersen/Barney vowels used in this work, the exemplars will consist of the number of zero crossings in a 512 byte section of the vowel, plus 3 coefficients derived from the DAUB4 wavelet transform of the same 512 byte section. The method used to capture the wavelet coefficients is the subject of this section.

Many signals which we may wish to analyze (including speech) are not defined only by frequency content, but also by the positions of one event with respect to another in time. This is common in the power generation industry where three phase generators produce output with a set spatial relationship. Many such examples can also be sighted in the medical profession, with one of the best being the output of heart monitoring devices. Wavelets, with their ability to focus on patterns in different frequency bands, are well suited for analyzing the positional relationships in these types of signals.

Since most of the relevant information is contained in the largest wavelet coefficients, speech exemplars can be extracted by using wavelets to record the relative positions of the largest wavelet coefficients in each frequency band. This approach must overcome two sources of variability to be successful. Firstly, phase differences in the vowels must be considered when performing the wavelet transforms. As mentioned earlier, the issue of phase correction has spawned some very complicated algorithms. These are required so that transformations performed on different vowels have a common reference point. Phase correction was not performed in the signal preparation stage because during the course of this work it became evident that it could be performed as part of the wavelet stage. This method capitalizes on the positional sensing capabilities of the wavelet in order to perform phase synchronization. In this implementation a series of ten 512 byte wavelet transforms were performed on the 1024 byte vowel buffer in an attempt to locate a reference point. This was done by using a moving window transform whose starting point incremented by 16 bytes each time. Figure 1 shows the positions (relative to time) of all wavelet coefficients in each frequency band after the transform is performed.

Figure 1

Band 1	Band 2	Band 3	Band 4	Band 5	Band 6	Band 7	Band 8	Band 9
0-1	2-3	4-7	8-15	16-31	32-63	64-127	128-255	256-511

In essence the moving window series yielded an array of 10 buffers of the type shown in Figure 1. These transformed buffers were then scanned to find the one having its maximum coefficient located at position 100 (or as close to 100 as possible) in frequency band 7. This provided a common reference point between all vowels. The choice of the band 7, while somewhat arbitrary, was due to its placement in the mid-frequency range and its acceptable temporal resolution. This would increase the chances of having frequency activity in this band for all vowels, thus providing a stable reference point.

Once one of the ten transformed buffers was selected, the exemplars for the vowel were chosen as the position of the maximum coefficients in bands 8 and 9 along with the sum of all the positions of maximum coefficients in all 9 bands. These three exemplars will be hitherto referred to as C1, C2, and C3 respectively. An example of actual data used in this process is shown in Figure 2. Again, these are the temporal locations of the maximum coefficients in each frequency band for the selected transform buffer.

Figure 2

Band 1	Band 2	Band 3	Band 4	Band 5	Band 6	Band 7	Band 8	Band 9
0	2	4	10	28	55	101	170	343

In this case we have:

C1 = 170, C2 = 343, and

C3 = 0+2+4+10+28+55+101+170+343=713.

The resulting exemplars C1, C2, and C3, are a snapshot in time of the behavior of the vowel relative to the mid-frequency band reference. Below, in Figure 3, a listing of the pseudo code has been provided to demonstrate more clearly how the exemplar extraction algorithm was implemented. Steps 1 through 6 were performed as part of the Macintosh exemplar extractor immediately after its initial signal preparation stage. Steps 7-13 are discussed in the next section.

Figure 3

(1) Place the preprocessed vowel in a 1024 byte buffer.

(2) Set the vowel buffer pointer to the beginning of the buffer.

(3) Perform a wavelet transform on the 512 byte section starting at the current pointer.

(4) Increment the buffer pointer by 16 bytes.

(5) If less than 10 transforms have been performed, go to step 3.

(6) Save the locations of all maximum coefficients for all ten of the wavelet transforms in a file along with the ten zero crossing counts.

(7) Locate frequency band 7 in each of the ten 512 byte transforms.

(8). Choose the transform whose maximum coefficient is placed closest to 100 in frequency band 7 (see Figure 2).

(9) Save the placement of all maximum coefficients in the other frequency bands associated with this particular transformation.

(10) Set C1 = the position of maximum coefficient in band 8.

(11) Set C2 = the position of maximum coefficient in band 9.

(12) Set C3 = the sum of the positions of maximum coefficients in all nine frequency bands.

(13) Set F0 = the zero crossing count which is the pitch estimate.

In addition to the phase correction issue, we also need to remember that the positions of maximum coefficients relative to reference band 7 will be dependent on the characteristic pitch of each individuals voice. Thus we should not expect to get similar values of C1, C2, and C3 for the same vowel unless both speakers have a similar pitch profile. This is because the entire speech signal is in essence modulated by this fundamental frequency causing shifting in the positions of coefficients among the different frequency bands. Thus, as noted in step 6 of Figure 3, each of the ten transforms are also saved with their respective pitch estimate (zero crossing count). We will see that this value, hitherto referred to as F0, will be critical to the success of the discrimination algorithm.

Figure 4 shows an actual block of ten sets of nine maximum wavelet coefficients along with the ten associated zero crossing counts (pitch estimate). The fifth set, in this case, would be selected for exemplar extraction because of the value of 101 in band 7. This set is also shown, minus the zero crossing value, in Figure 2. Figure 4, which is shown in the file format written to by the Macintosh feature extractor (Figure 3, Step 6), contains all the information required to determine C1, C2, C3, and F0. We will see in the next section that this four part exemplar will be sufficient to distinguish the nine vowels one from another.

Figure 4

Band 1	Band 2	Band 3	Band 4	Band 5	Band 6	Band 7	Band 8	Band 9	Pitch
0	2	4	12	22	57	117	202	407	56
1	2	6	12	30	61	113	194	391	61
0	2	4	11	30	59	109	186	375	64
1	2	5	11	29	57	105	178	359	65
0	2	4	10	28	55	101	170	343	59
1	3	5	10	27	62	97	162	511	62
0	2	7	14	26	51	123	154	311	59
1	2	5	13	25	49	119	146	295	58
0	3	7	13	24	62	115	138	279	59
0	2	4	12	23	62	111	130	263	60

Petersen/Barney Vowel Discrimination

As mentioned in the previous section, the Macintosh exemplar extractor generates an output file containing data of the form shown in Figure 4. Data of this type is stored for each spoken instance of each vowel in the Petersen/Barney set. The Macintosh program does not, however, perform steps 7 through 13 shown in the pseudo code listing of Figure 3. These steps were performed manually due to the fact that: (1) the discrimination system was only available on a Sun SPARC 10 thus requiring the data to be transferred via disk; and (2) the act of scanning all ten transformations and selecting the one with a maximum near position 100 (band 7) is a simple operation having no implications on the viability of this work. Thus the reader should assume that all sets of C1, C2, C3, and F0 presented to the discriminator were derived by using this hybrid scheme.

Before discussing the actual discrimination system, it is important to note that with the Petersen/Barney vowels we gain a major advantage in knowing the pitch. By definition, these vowels are fairly localized in frequency space with respect to each other. This means that before even using C1, C2, or C3 in the discriminator, we can break the vowels down into smaller groups according to their zero crossing counts (F0). The vowels shown in Figure 0 were found to reside in three major pitch groups. Group A, which contains vowels 1 through 4, were found to have zero crossings above 50 counts per 512 bytes.

Group B, which contains vowels 5, 6, and 9, on average had counts between 30 and 50 per 512 bytes. Group C contains vowels 7 and 8 and generally had zero crossings below 30 counts per 512 bytes. The immediate advantage of this technique is that each vowel can be classified as having membership in one of three groups before the discrimination algorithm is even applied. This has the net effect of increasing the discrimination rate because of the reduction in the number of vowels that must be separated one from another.

The discriminant analysis system used in this project was the SAS DISCRIM Procedure [15] running on a Sun Microsystems SPARC 10. When presented with the input, the DISCRIM Procedure generates quadratic discriminant functions for classifying observations into two or more groups based on exemplars containing one or more variables. The resulting discriminant function is derived from the generalized pairwise squared distance and would be easily ported to other computer systems. In this case the prior probabilities of occurrence are assumed to be equal for all inputs. Upon completion, SAS produces a summary file showing implementation details and the percentage of time the vowels were properly correlated to their number (1 through 9) based on their exemplars. The vowel exemplar input and resulting SAS output are shown in Appendix A for group A, Appendix B for group B, and Appendix C for group C.

While it is possible to run the SAS discriminator directly on the four part exemplars shown in the appendices, it is generally advantageous to create a new set of inputs based on some intuition about the data. In this case, a set of 16 new C and F exemplars were created inside the SAS program after reading C1, C2, C3, and F0 from disk. The expanded set of exemplars were generated based on the knowledge that there were relationships between these values that could be expressed as ratios. These ratios, shown below in Figure 5, were used to accentuate the interdependencies between pitch and the temporal placement of wavelet coefficients. Note that some exemplar values have been redefined, but still retain the influence of the original value. All vowel discrimination was performed using this new set of exemplars.

Figure 5

Calculate First	Calculate Second
C7 = (C2 * C1)/C3	F0 = log10(F0)
C6 = (C1 * C3)/C2	F1 = C0/F0
C5 = (C2 + C1)/C3	F2 = C1/F0
C4 = (C2 - C1)/C3	F3 = C2/F0
C3 = C3	F4 = C4/F0
C0 = (C1 + C2 + C3)/C3	F5 = C5/F0
C1 = C3/(C3 + C1)	F6 = C6/F0
C2 = C3/(C3 + C2)	F7 = C7/F0

After running the SAS DISCRIM procedure on the expanded exemplar data, the goal of a 75% minimum discrimination rate was achieved. These results, which are shown in Figure 6, are based on twelve repetitions of the vowels; one from each of the twelve different speakers.

Figure 6

Vowel # (derived from)	Discrimination Rate
1 (EAT)	83.33 %
2 (IT)	75.00 %
3 (ANY)	75.00 %
4 (AT)	75.00 %
5 (UP)	83.33 %
6 (AMISH)	91.67 %
7 (OOMPH)	91.67 %
8 (OOZE)	83.33 %
9 (AUTO)	75.00 %

The encouraging part of these results is that reasonable discrimination rates were achieved in spite of the emphasis on conserving CPU cycles. This indicates that much better rates could be achieved with signal processing and wavelet algorithms that are optimized for this task. Of course, the penalty would be increased pressure on CPU execution time.

Summary

In general, the results of this work were quite encouraging. In addition to achieving the minimum goal of a 75% discrimination rate, the following observations were made:

- The ability of the wavelet transform to detect the placement events in time can be used to synchronize the phases of multiple signals by looking for predetermined signatures.
- After phase synchronization, it is also possible to distinguish one vowel from another based on the localization of events in time.
- A pitch estimation algorithm can be implemented by simply calculating the number of zero crossings in the utterance.
- For speech with specific frequency localization (such as the Petersen/Barney vowels), the pitch estimate can be used to subdivide the speech prior to the discrimination algorithm.
- A simple discrimination system, employing the techniques described herein, could be implemented to respond to single vowel commands.
- The wavelet phase detection algorithm could be used to look for the start of an event in any signal (speech, image, medical, 3 phase power, etc.), by simply searching for the placement (in time) of a given wavelet coefficient according to a predetermined specification.

- Where CPU power is available, a continuous moving window wavelet transform could be used to differentiate vowels in real-time. This would involve a continuous comparison of the positions of the wavelet coefficients in each frequency band after each transform.
- In light of the success achieved with the simplified pitch estimation and wavelet algorithms, much better results could be expected with more complex algorithms and increased computing power.

Appendix A

Group A 0
10:38 Saturday, February 4, 1995

S#	F0	C1	C2	C3
1	50	247	495	955
1	51	175	357	726
1	59	170	343	713
1	65	251	507	950
1	72	207	417	834
1	75	209	456	860
1	79	160	491	863
1	87	174	348	714
1	89	157	315	658
1	92	170	438	828
1	102	153	366	730
1	113	138	269	624

S#	F0	C1	C2	C3
2	50	160	511	889
2	50	248	408	857
2	51	204	404	814
2	51	189	374	756
2	56	212	413	830
2	62	215	431	844
2	65	204	428	842
2	67	212	426	832
2	71	233	466	908
2	74	243	329	763
2	78	246	495	945
2	81	248	497	937

S#	F0	C1	C2	C3
3	56	207	472	859
3	58	181	452	826
3	66	205	416	820
3	72	133	511	851
3	75	213	429	829
3	76	240	483	924
3	85	250	400	843
3	86	159	324	671
3	88	240	486	932
3	89	192	386	764
3	97	250	501	954
3	103	183	424	818

S#	F0	C1	C2	C3
4	50	194	274	666
4	50	199	400	796
4	50	197	393	774
4	54	199	400	788
4	61	200	400	793
4	62	252	418	872
4	64	254	419	883
4	64	202	403	797
4	66	199	400	807
4	70	207	415	808
4	75	202	406	817
4	95	152	366	707

Exemplars presented as input to SAS for vowels 1 through 4.
Vowels were derived from 12 individual speakers.

Discriminant Analysis

48 Observations	47 DF Total
16 Variables	44 DF Within Classes
4 Classes	3 DF Between Classes

Class Level Information

SOUND	Frequency	Weight	Proportion	Prior Probability
1	12	12.0000	0.250000	0.250000
2	12	12.0000	0.250000	0.250000
3	12	12.0000	0.250000	0.250000
4	12	12.0000	0.250000	0.250000

Discriminant Analysis Pooled Covariance Matrix Information

Covariance Matrix Rank	Natural Log of the Determinant of the Covariance Matrix
16	-148.15458

Discriminant Analysis

Pairwise Generalized Squared Distances Between Groups

$$D^2(i|j) = (\bar{X}_i - \bar{X}_j)' \, COV^{-1} \, (\bar{X}_i - \bar{X}_j)$$

Generalized Squared Distance to SOUND

From SOUND	1	2	3	4
1	0	7.44579	7.15240	9.40454
2	7.44579	0	6.07952	2.16032
3	7.15240	6.07952	0	6.39367
4	9.40454	2.16032	6.39367	0

Discriminant Analysis Linear Discriminant Function

$$\text{Constant} = -.5\ \bar{X}_j'\ COV^{-1}\ \bar{X}_j \qquad \text{Coefficient Vector} = COV^{-1}\ \bar{X}_j$$

SOUND

	1	2	3	4
CONSTANT	-28387112	-28378857	-28375796	-28373005
C0	-76226930	-76224439	-76203559	-76214671
C1	-3034464	-3035846	-3045365	-3016966
C2	381278534	381282879	381180965	381216076
C3	-3759	-3764	-3757	-3763
C4	85470437	85476876	85451823	85456237
C6	44983	45024	44971	45020
C5	-65617659	-65625900	-65597579	-65616804
C7	-3916	-3916	-3915	-3915
F0	40796	40594	40970	40500
F1	143298970	143292692	143253258	143273612
F2	79965435	79963699	79968494	79922862
F3	-702184950	-702195605	-701994757	-702070326
F4	-188766574	-188777425	-188722670	-188736005
F5	148777224	148788222	148732455	148767770
F6	8883	8893	8880	8891
F7	-70022	-70099	-70000	-70092

Discriminant Analysis

Classification Summary for Calibration Data: WORK.AA

Resubstitution Summary using Linear Discriminant Function

Generalized Squared Distance Function:

$$D_j^2(X) = (X-\bar{X}_j)' \, COV^{-1} \, (X-\bar{X}_j)$$

Posterior Probability of Membership in each SOUND:

$$Pr(j|X) = \exp(-.5\, D_j^2(X)) \,/\, SUM_k \exp(-.5\, D_k^2(X))$$

Number of Observations and Percent Classified into SOUND:

From SOUND	1	2	3
1	10 83.33	1 8.33	1 8.33
2	0 0.00	9 75.00	1 8.33
3	1 8.33	1 8.33	9 75.00
4	0 0.00	2 16.67	1 8.33
Total Percent	11 22.92	13 27.08	12 25.00
Priors	0.2500	0.2500	0.2500

Discriminant Analysis

Classification Summary for Calibration Data: WORK.AA

Resubstitution Summary using Linear Discriminant Function

Number of Observations and Percent Classified into SOUND:

From SOUND	4	Total
1	0 0.00	12 100.00
2	2 16.67	12 100.00
3	1 8.33	12 100.00
4	9 75.00	12 100.00
Total	12	48
Percent	25.00	100.00
Priors	0.2500	

Error Count Estimates for SOUND:

	1	2	3	4	Total
Rate	0.1667	0.2500	0.2500	0.2500	0.2292
Priors	0.2500	0.2500	0.2500	0.2500	

Appendix B

Group B 0
09:53 Saturday, February 4, 1995

S#	F0	C1	C2	C3	S#	F0	C1	C2	C3	S#	F0	C1	C2	C3
5	41	136	511	852	6	49	166	511	855	9	45	255	511	961
5	35	246	492	947	6	39	192	511	903	9	40	166	511	866
5	49	130	258	560	6	37	255	511	972	9	30	205	511	919
5	45	135	365	675	6	47	255	511	965	9	43	234	460	910
5	48	232	467	896	6	40	186	446	836	9	35	236	479	909
5	39	212	415	828	6	45	199	511	901	9	33	208	418	813
5	45	234	464	895	6	49	172	511	859	9	49	231	465	895
5	42	197	395	774	6	47	255	450	858	9	35	173	353	737
5	30	219	511	933	6	49	251	511	952	9	35	243	511	952
5	35	226	454	855	6	42	255	511	968	9	37	209	499	903
5	41	199	400	777	6	44	201	511	901	9	32	255	511	947
5	41	238	295	731	6	44	165	501	860	9	44	202	405	807

Exemplars presented as input to SAS for vowels 5, 6, and 9.
Vowels were derived from 12 individual speakers.

Discriminant Analysis

36 Observations	35 DF Total
16 Variables	33 DF Within Classes
3 Classes	2 DF Between Classes

Class Level Information

SOUND	Frequency	Weight	Proportion	Prior Probability
5	12	12.0000	0.333333	0.333333
6	12	12.0000	0.333333	0.333333
9	12	12.0000	0.333333	0.333333

Discriminant Analysis Pooled Covariance Matrix Information

Covariance Matrix Rank	Natural Log of the Determinant of the Covariance Matrix
16	-154.26124

Discriminant Analysis

Pairwise Generalized Squared Distances Between Groups

$$D^2(i|j) = (\bar{X}_i - \bar{X}_j)' \, COV^{-1} \, (\bar{X}_i - \bar{X}_j)$$

Generalized Squared Distance to SOUND

From SOUND	5	6	9
5	0	10.31077	7.47494
6	10.31077	0	6.24642
9	7.47494	6.24642	0

Discriminant Analysis Linear Discriminant Function

$$\text{Constant} = -.5\ \bar{X}_j' \ COV^{-1}\ \bar{X}_j \qquad \text{Coefficient Vector} = COV^{-1}\ \bar{X}_j$$

SOUND

	5	6	9
CONSTANT	-32277681	-32281120	-32272681
C0	-125348089	-125214528	-125169474
C1	684614944	684658333	684466760
C2	-190942609	-191604568	-191613466
C3	-16645	-16619	-16604
C4	-248049881	-248228165	-248182120
C6	140905	140715	140610
C5	-71428008	-71271549	-71227840
C7	-6928	-6928	-6927
P0	45865	46073	46501
P1	176836947	176621691	176550619
P2	-1.01563E9	-1.01571E9	-1.01541E9
P3	412291135	413378023	413384539
P4	385210673	385506957	385436056
P5	116039107	115786043	115712962
P6	27172	27129	27105
P7	-188962	-188651	-188483

Discriminant Analysis

Classification Summary for Calibration Data: WORK.AA

Resubstitution Summary using Linear Discriminant Function

Generalized Squared Distance Function:

$$D_j^2(X) = (X-\bar{X}_j)' \, COV^{-1} \, (X-\bar{X}_j)$$

Posterior Probability of Membership in each SOUND:

$$Pr(j|X) = \exp(-.5\, D_j^2(X)) \,/\, SUM_k \exp(-.5\, D_k^2(X))$$

Number of Observations and Percent Classified into SOUND:

From SOUND	5	6	9	Total
5	10	0	2	12
	83.33	0.00	16.67	100.00
6	0	11	1	12
	0.00	91.67	8.33	100.00
9	1	2	9	12
	8.33	16.67	75.00	100.00
Total	11	13	12	36
Percent	30.56	36.11	33.33	100.00
Priors	0.3333	0.3333	0.3333	

Error Count Estimates for SOUND:

	5	6	9	Total
Rate	0.1667	0.0833	0.2500	0.1667
Priors	0.3333	0.3333	0.3333	

Appendix C

Group C 0
09:52 Saturday, February 4, 1995

S#	F0	C1	C2	C3	S#	F0	C1	C2	C3
7	28	255	511	974	8	20	162	314	699
7	23	255	511	960	8	13	255	511	983
7	29	157	511	869	8	19	255	511	968
7	23	159	316	673	8	28	250	258	710
7	29	238	511	946	8	19	235	462	880
7	21	202	511	916	8	25	215	414	830
7	29	255	511	982	8	21	255	511	954
7	29	165	492	849	8	21	209	511	921
7	19	255	511	976	8	18	219	511	908
7	27	227	351	782	8	29	180	287	658
7	20	255	511	954	8	21	207	511	907
7	26	204	511	925	8	29	169	511	877

Exemplars presented as input to SAS for vowels 7 and 8.
Vowels were derived from 12 individual speakers.

Discriminant Analysis

24 Observations	23 DF Total
16 Variables	22 DF Within Classes
2 Classes	1 DF Between Classes

Class Level Information

SOUND	Frequency	Weight	Proportion	Prior Probability
7	12	12.0000	0.500000	0.500000
8	12	12.0000	0.500000	0.500000

Discriminant Analysis Pooled Covariance Matrix Information

Covariance Matrix Rank	Natural Log of the Determinant of the Covariance Matrix
16	-149.7567

Discriminant Analysis

Pairwise Generalized Squared Distances Between Groups

$$D^2(i|j) = (\bar{X}_i - \bar{X}_j)' \, COV^{-1} \, (\bar{X}_i - \bar{X}_j)$$

Generalized Squared Distance to SOUND

From SOUND	7	8
7	0	4.73158
8	4.73158	0

Discriminant Analysis Linear Discriminant Function

$$\text{Constant} = -.5\ \bar{X}_j'\ COV^{-1}\ \bar{X}_j \qquad \text{Coefficient Vector} = COV^{-1}\ \bar{X}_j$$

SOUND

	7	8
CONSTANT	-116920497	-116919457
C0	-178032482	-178116795
C1	2191695849	2191836091
C2	-1.33626E9	-1.33607E9
C3	-82773	-82780
C4	-917554648	-917551644
C6	696802	696867
C5	-89169658	-89244292
C7	-18294	-18296
P0	-1264658	-1264670
P1	174968199	175073034
P2	-2.84637E9	-2.84657E9
P3	2241564904	2241342867
P4	1273630932	1273634051
P5	129206960	129298509
P6	114369	114379
P7	-869557	-869635

Discriminant Analysis

Classification Summary for Calibration Data: WORK.AA

Resubstitution Summary using Linear Discriminant Function

Generalized Squared Distance Function:

$$D_j^2(X) = (X-\bar{X}_j)' \, COV^{-1} \, (X-\bar{X}_j)$$

Posterior Probability of Membership in each SOUND:

$$Pr(j|X) = \exp(-.5\, D_j^2(X)) \,/\, SUM_k \exp(-.5\, D_k^2(X))$$

Number of Observations and Percent Classified into SOUND:

From SOUND	7	8	Total
7	11	1	12
	91.67	8.33	100.00
8	2	10	12
	16.67	83.33	100.00
Total	13	11	24
Percent	54.17	45.83	100.00
Priors	0.5000	0.5000	

Error Count Estimates for SOUND:

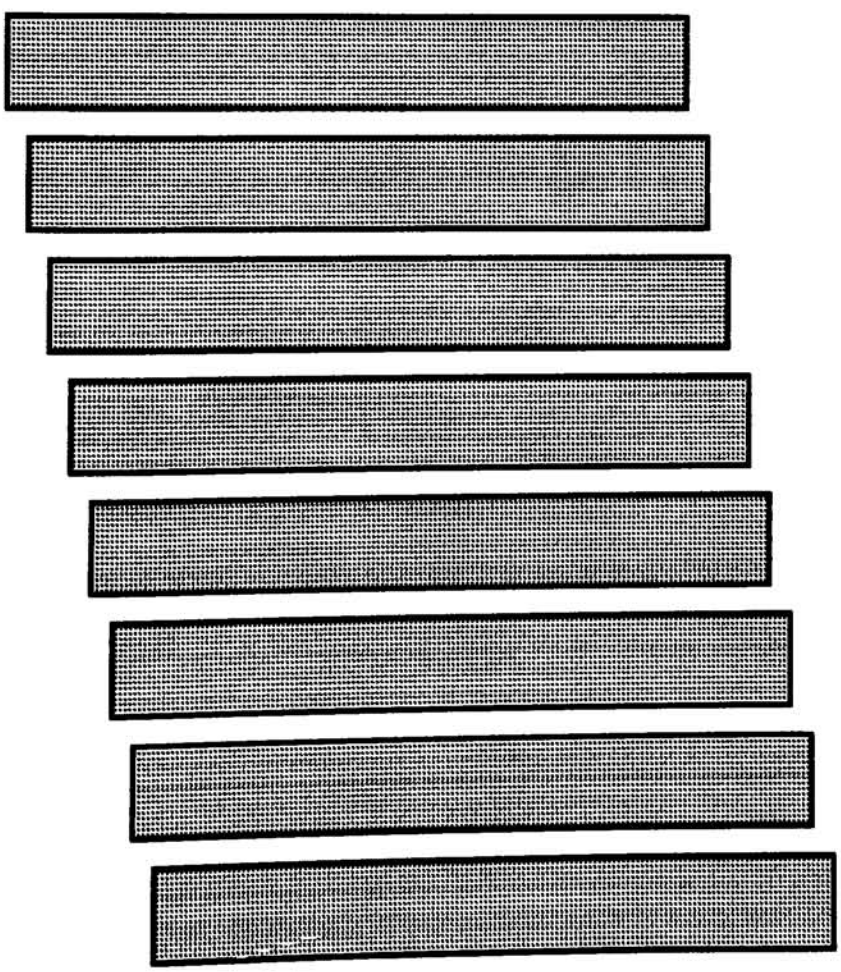
	7	8	Total
Rate	0.0833	0.1667	0.1250
Priors	0.5000	0.5000	

Appendix D

Wavelet Based Phase Alignment

Speech Signal

Sampled Speech Buffer (1024 bytes)



Moving Window Wavelet Series (16 byte increments/512 byte window)

REFERENCES

1. Speech Frequently Asked Questions (FAQ); Andrew Hunt, comp.speech on the internet, 1994.

2. Fundamentals of Speech Recognition; Lawrence Rabiner & Biing-Hwang Juang, Englewood Cliffs NJ: PTR Prentice Hall (Signal Processing Series), c1993 ISBN 0-13-015157-2.

3. Speech recognition by machine; W. A. Ainsworth, London: Peregrinus for the Institute for Electrical Engineers, c1988.

4. Speech synthesis and recognition; J. N. Holmes, Wokingham: Van Nostrand Reinhold, c1988.

5. Speech Communication: Human and Machine; Douglas O'Shaughnessy, Addison Wesley series in Electrical Engineering (Digital Signal Processing), 1987.

6. Electronic speech recognition: techniques; technology and applications edited by Geoff Bristow, London: Collins, 1986.

7. Readings in Speech Recognition; edited by Alex Waibel & Kai-Fu Lee, San Mateo: Morgan Kaufmann, c1990.

8. Signal and Image Processing with Neural Networks; Timothy Masters, John Wiley & Sons, c1994 ISBN 0-471-04963-8.

9. Numerical Recipes in C (The art of scientific computing); William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery, Cambridge: Cambridge University Press, c1992 ISBN 0-521-43108-5.

10. Numerical Recipes (Example Book [C]); William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery, Cambridge: Cambridge University Press, c1992 ISBN 0-521-43720-2.

11. Inside Macintosh (Volumes I, II, and II); Apple Computer, Addison-Wesley, c1985 ISBN 0-201-17737-4.

12. Inside Macintosh (Volume VI); Apple Computer, Addison-Wesley, c1991 ISBN 0-201-57755-0.

13. Visual Representations of Speech Signals; edited by Martin Cooke, Steve Beet, and Malcolm Crawford, John Wiley & Sons, c1993 ISBN 0-471-93537-9.

14. Desktop SPARC Hardware Owners Guide, 1992 Sun Microsystems, Inc.

15. SAS Institute Inc. SAS User's Guide: Statistics, Version 5 Edition. Cary, NC: SAS Institute Inc. 1985. 956 pp.

16. Neural Network Adaptive Wavelets for Signal Representation and Classification; Harold H. Szu, Brian Telfer, and Shubha Kadambe, Optical Engineering, September 1992.

17. Applications of Adaptive Wavelets for Speech; Shubha Kadambe and Pramila Srinivasan, Optical Engineering, July 1994.

18. Wave of the future; Hans Christian Von Baeyer; Discover Magazine, May 1995.